

# Unity入门

童伟华

2019-04-26

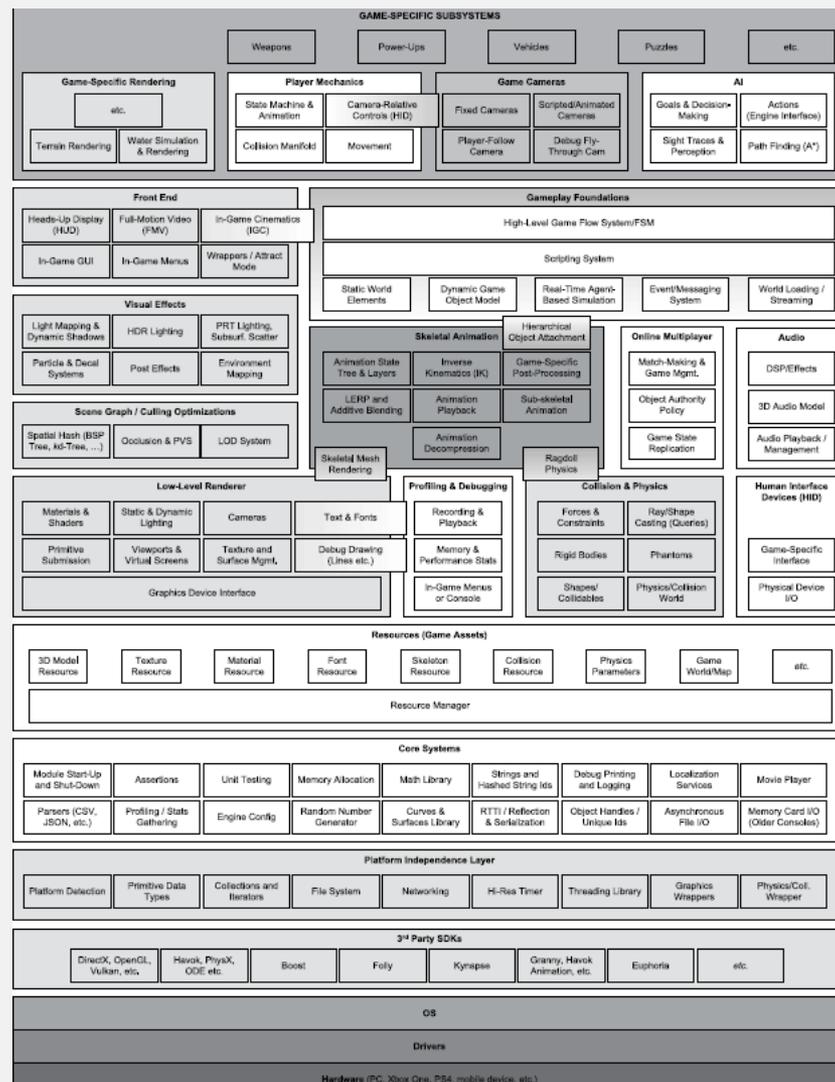
# 游戏引擎介绍

- 什么是游戏引擎？
  - 游戏引擎好比赛车的引擎，**是用于控制所有游戏功能的主程序**，从计算碰撞、物理加速系统和物体的相对位置，到接受玩家的输入，以及按照正确的音量输出声音等等
  - 类似题材的游戏总是有些相同的代码，**代码复用可以大大减少游戏的开发周期和开发费用**，慢慢地这些通用代码就形成了引擎的雏形，伴随技术的发展，最终演变成今天的游戏引擎

# 游戏引擎的架构

## • 游戏引擎的主要组件

- 渲染引擎
- 物理引擎
- 碰撞检测引擎
- 音效引擎
- 脚本引擎
- 动画引擎
- 人工智能引擎
- 网络引擎
- 场景管理引擎
- IO管理引擎
- ...



# 游戏引擎介绍

- 现有的一些著名商业游戏引擎
  - EpicGames' [Unreal Engine4](#) (已创造了超过100亿美元的收, 游戏开发的高端市场)
  - Valve's Source engine
  - Crytek's CRYENGINE® 3
  - Electronic Arts DICE's Frostbite™ engine
  - [Unity](#) (跨平台, 用户数量多, 容易上手)
  - ...
- 一些流行的开源游戏引擎
  - Irrlicht Engine
  - Ogre Engine

# Unity介绍

- 商业游戏引擎
- 在手机游戏平台，Unity已经处于霸主地位，70%的占有率
- 版本：个人版（免费）、加强版、专业版
- <https://unity.com/cn>
- 开发跨平台：Windows和macOS
- 发布跨平台：Windows, macOS, Linux, Android, PS4, XBOXONE, Switch, WebGL...等等

# Unity介绍

- 主要特点
  - 可视化开发界面
  - C#脚本语言
  - 模块化组件
  - 底层代码C++编写，C#作为游戏脚本语言，利用Mono实现跨平台特性
- 资源商店
  - 插件
  - 模型
  - ...

# Unity介绍

- 学习资料

- Unity自带教程
- Joseph Hocking, 蔡俊鸿译, Unity 5 实战, 清华大学出版社 (最新英文版, Unity in Action, 2ed Ed., Manning Publications.)
- 宣雨松, Unity 3D 游戏开发, 第二版, 人民邮电出版社
- 冯乐乐, Unity Shader 入门精要, 人民邮电出版社
- 网络视频

# Unity初识

- Unity: 所见即所得 (What You See Is What You Get, ab., WYSIWYG), 编辑器与游戏引擎融合在一起
- 基本概念
  - 游戏项目 (Project) 有若干游戏场景组件
  - 游戏场景由若干游戏对象组成
  - 游戏对象由若干游戏资源组成

# Unity初识

- Project视图：可以管理所有的游戏资源  
（譬如脚本，着色器，场景，材质，动画控制器等，可分为外部资源，内部资源）
- Hierarchy视图：管理所有游戏对象（负责引用外部资源，通过程序来控制其行为，譬如3D物体，2D物体，特效，光，视频，音频，摄像机等等）
- Inspector视图：承担所有游戏对象即游戏资源组件参数的编辑工作

# Unity初识

- Scene视图：游戏场景的自由视角，主要负责安排游戏对象的摆放
- Game视图：显示游戏的主摄像机看到的内容，即游戏最终展示给玩家的画面

# Example 1: Hello World!

- (1) 创建新项目
- (2) 创建新的C#脚本
- (3) 创建新的GameObject
- (4) 将脚本拖动到对象上
- (5) 给脚本添加日志命令
- (6) 单击Play按钮

# 场景导航和键盘快捷键

表 A-1 不同类型的鼠标按键的场景导航控件

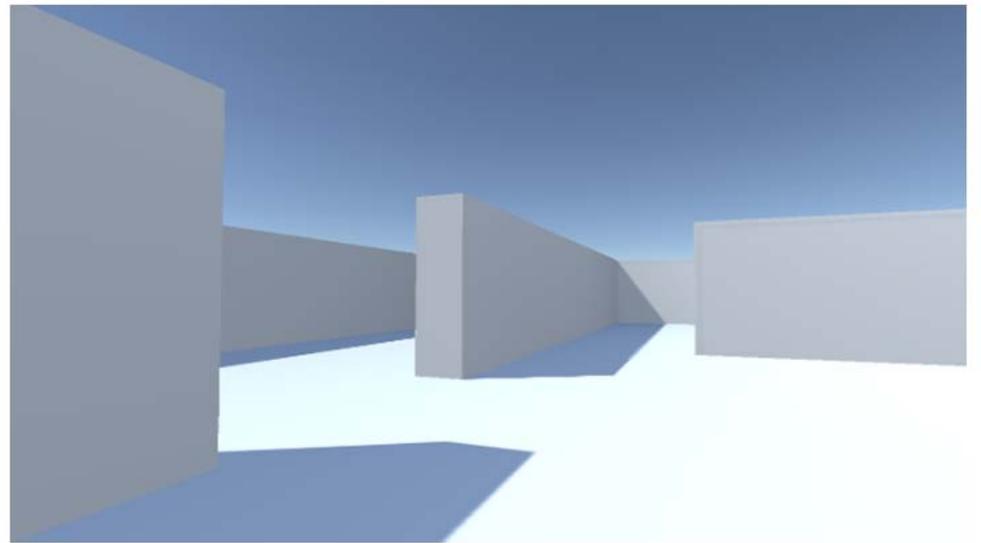
导航行为	三个按键的鼠标	两个按键的鼠标	一个按键的鼠标
Move	中键单击	Alt + Command + 左键/拖动	Alt + Command + 单击/拖动
Orbit	按住 Alt + 左键/拖动	Alt + 左键/拖动	Alt + 单击/拖动
Zoom	按住 Alt + 右键/拖动	Alt + 右键/拖动	Alt + Ctrl + 单击/拖动

表 A-2 有用的键盘快捷键

按 键	功 能
W	平移(移动选中的对象)
E	旋转(旋转选中的对象)
R	缩放(改变选中对象的大小)
T	矩形工具(操作 2D 对象)
F	将视野聚焦在选中的对象上
V	对齐到顶点
Ctrl/Command+Shift+N	新 GameObject
Ctrl/Command+P	运行游戏
Ctrl/Command+R	刷新对象
Ctrl/Command+1	设置当前窗口为 Scene 视图
Ctrl/Command+2	设置为 Game 视图
Ctrl/Command+3	设置为 Inspector 视图
Ctrl/Command+4	设置为 Hierarchy 视图
Ctrl/Command+5	设置为 Project 视图
Ctrl/Command+6	设置为 Animation 视图

# Example 2: 置身3D空间

- Unity的可视化编辑器允许用户不需要编写一些模板代码就能快速构建3D游戏原型
- (1) 在场景中放置一个玩家
- (2) 编写移动对象的脚本
- (3) 实现FPS控制



# Example 3: 游戏中的美术资源

- 如何提升游戏的视觉效果？
- 美术资源
  - 2D图像：纹理
  - 3D模型
  - 材质：金属、镜子等
  - 动画
  - 粒子：用于创建并控制大量小型对象的规则机制，譬如水、火、雾、布料、毛发等

# Example 4: 在游戏中放置GUI

- 每个游戏除了进行虚拟场景外，还需要一些抽象交互，显示一些信息，通过GUI完成
  - (1) 规划界面
  - (2) 设计显示界面中的UI元素
  - (3) 编写与UI元素的交互
  - (4) 编写响应GUI事件的代码
  - (5) 使场景响应GUI上的动作

# Example 5: 玩家移动和动画

- 第三人称游戏：摄像机在玩家角色外部，并俯视这个角色
  - (1) 将角色模型导入到场景
  - (2) 实现摄像机控制，以观察角色
  - (3) 编写脚本，让玩家能够在地面上跑
  - (4) 给移动脚本添加跳跃功能
  - (5) 基于移动播放模型的动画

# Example 6: 在游戏中添加交互设施和物件

- 游戏中的交互设施，譬如门、机关等
- 游戏状态，譬如玩家的当前状态，收集的物件等
- (1) 创建门和其他设施
- (2) 通过碰撞与对象交互
- (3) 管理仓库数据和游戏状态
- (4) 使用和装备物品的仓库UI

# Example 7: 播放音效和音乐

- 播放背景音乐和音效
  - (1) 导入音效的音频文件
  - (2) 为敌人和设计播放音效
  - (3) 编写一个音频管理器控制音量
  - (4) 优化音乐的加载
  - (5) 控制音乐和音效的音量

# Example 8: 将游戏接到互联网

- 互联网游戏：大型，多人，交互
- 网络通讯，服务器搭建，协同，分布式数据库等

# Example 9: 将各部分整合为一个完整的游戏

- 游戏的外围结构，包括关卡的切换，游戏的结束，进度保存等

# Example 10: 将游戏部署到玩家的设备

- 跨平台：在Unity环境中开发，可部署到各个游戏平台
- 依据平台，构建游戏